AMENDMENTS TO THE SPECIFICATION

Please amend the Specification as follows:

Amend the paragraph on page 12, lines 4-12, as follows:

The planner 105 is typically implemented as code or firmware and is typically a conventional type of planner. Similarly, the executor 110 is typically implemented as code or firmware and is typically a conventional type of executor. The particular implementation of the planner 105 and executor [[105]] 110 is typically dependent on the available operating system support, disk array support, and disk support for a data system that will perform migration planning and execution.

Amend the paragraph on page 14, line 17-page 15, line 6, as follows:

An embodiment of the method 200 first modifies the input to the planner 105 in order to treat the data chunk(s) that are being moved, as if each chunk is currently located at both the source (old) location and destination (new) location. The modified input to the planner 105 generates a plan 130 in order to achieve the target configuration 120. The method 200 then post-processes the plan to conform to the rules (constraints) of the migration. An example of a migration rule is that each device is to be only involved in a single transfer at a time by excluding moves which use devices that are ~~that are~~ currently involved in a transfer. Another example of a migration rule is that each device can be the source of at most two moves and the destination of at most one move.

Amend the paragraph on page 24, line 15-page 25, line 14, as follows:

Figure 6 is a block diagram illustrating an apparatus 600 for adaptive migration planning and execution, in accordance with another embodiment of the invention. The apparatus 600 includes components as similarly described for apparatus 100 in Figure 2. In addition, the apparatus [[500]] 600 includes a background load estimator 605 that can compute the background load on the storage system from the application. The background load estimator 605 may be, for example, a known host array instrumentation that detects the processor, disk, and application utilization. As another example, the background load estimator 605 may be an instrumentation that estimates the background load based upon interference. The instrumentation measures the delay amount during a move execution to determine the background load. For

example, the access speed for a disk or array in a system is known, and the size of the data chunk is also known. If a move is known to require about 10 seconds to complete in a system, but the instrumentation measures about 20 seconds to actually complete the move, then there is a 100% application load. If the instrumentation measures about 15 seconds to actually complete the move, then there is a 50% application load. If the instrumentation measures about 30 seconds to actually complete the move, then there is a 200% application load.

Amend the paragraph on page 25, line 23-page 26, line 17, as follows:

Figure 7A is a flowchart illustrating a method <u>700</u> for adaptive migration planning and execution, in accordance with another embodiment of the invention. It is further noted that the steps in Figure 7A, as similarly shown in Figure 3A, are performed as described above. The flowchart of Figure 7A illustrates an option for handling background load information. In step 705, the application load information or background load information is estimated, and this estimated load information is then fed into the planner 105. This estimated load information is considered by the planner 105 when determining a modified plan 130' in step 205 in Figure 7A. As an example, the in-progress moves 225 can be treated as part of the background load. For example, if the original background load is about 25% and a migration move involving a write will add another 50% to the background load, then the background load will become 75% utilized, and this estimated load information is fed into the planner 105 prior to generating a modified plan 130'.